2006 CCRTS
The State of the Art and the State of the Practice

Testing Agile Information Management Systems with Video Test Client
Case Study - DIMES

Lok Kwong Yan
Air Force Research Laboratory / IFTC
525 Brooks Road, Rome, NY 13441

Tel: (315) 330-2756 Fax: (315) 330-2953
Email: Lok.Kwong.Yan@rl.af.mil

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **JUN 2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE **Testing Agile Information Management Systems with Video Test Client. Case Study -DIMES**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Air Force Research Laboratory,AFRL/IFTC,525 Brooks Road,Rome,NY,13441**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**41**	

Testing Agile Information Management Systems with Video Test Client Case Study - DIMES

Lok Kwong Yan
Air Force Research Laboratory / IFTC
525 Brooks Road, Rome, NY 13441

Abstract:

In a Network Centric Warfare (NCW) environment, geographically distributed forces are networked together to attain military advantage, e.g. force agility. At the core of NCW is the ability for these distributed forces to collaborate and share information, which are the two services provided by information management systems located on the Global Information Grid (GIG). Due to the fact that a single system can not be perfect for all possible applications, system architects must sort through the different choices and select the best one for the desired application. In order to do so, designers must be able to understand what the capabilities of the system are in a common language. Quality of Service (QoS) metrics provide one means of describing these capabilities.

Like netcentric forces, it is desired that the supporting information management system also be agile. The notion of an agile information management system (AIMS) is introduced and the associated QoS attributes are defined. Video Test Client (VTC) is a QoS test platform that uses audio and video data as the basic unit of information. It is capable of using both pre-recorded and live video, i.e. video telecasting and teleconferencing, steams for testing. Its design and capabilities are also discussed.

VTC was used to obtain latency and jitter measurements for the Distributed Information Management Enterprise Service (DIMES), an AIMS under development at the Air Force Research Laboratory, Information Directorate. Preliminary results shows that DIMES introduces 787µs of latency, on average, resulting in a maximum of 190 good quality simultaneous collaboration streams with 20fps audio and 30fps video.

1 Introduction

Network Centric Warfare (NCW) seeks to leverage the combat power that can be realized by networking distributed forces. Once networked, these forces can collaborate and share their information to attain shared battlespace awareness, force agility, synchrony, etc. The quality of the end result relies heavily on the effectiveness of the information management system used to provide the collaboration and information sharing services to the edge users. It is therefore necessary to have a means of describing an information management system's capabilities so that the appropriate system can be employed given an application.

One common way of describing the capabilities of a system is through its Quality of Service (QoS) attributes. QoS is a term traditionally used to describe the performance capabilities of communication protocols and networks. It has since been adapted to describe technologies above and beyond the Open Systems Interconnection (OSI) network and transport layers and expanded to include non-performance related metrics, e.g. security [1], [2]. In the broadest sense, a QoS attribute can be any desirable characteristic (quality) of a specific service. Thus, QoS attributes can be used to describe any system. The choice and definition of these characteristics are highly dependent on the specific type of system under test and the intended user domain. The type of system under consideration is agile information management systems (AIMS) and the intended user domain is agile forces. The concept of agile information management systems, systems that support agile forces, is described in section 2.

Information management systems are designed to securely provide the right information at the right time to all of its users. The "right information" and the "right time" characteristics of the information are subjective and are defined by the user. The ability of the information management system to satisfy the user's description of what the right information is depends on the richness of the information management system's brokering mechanism. Finally, the information management system's ability to provide information at the right time depends on the system's QoS attributes. The specific qualities that are applicable to AIMS are discussed in section 3.

Video Test Client (VTC) is a flexible test platform that provides both quantitative and qualitative measurement results. By using audio and video frames as the basic units of information, users are able to attain a qualitative feel of what the system under test is capable of. The test platform is capable of using pre-recorded or live audio and video channels for testing. By doing so, the system can be tested under real world conditions using a reasonable application, audio/video collaboration. Furthermore, VTC can be used to assess the fitness of an information management system not only for information sharing but also for audio/video collaboration. Its design and capabilities are presented in section 4.

The Distributed Information Management Enterprise Service (DIMES) is an information management system under development at the Air Force Research Laboratory, Information Directorate. It is based on the Information Management Enterprise Service (IMES) which is also under development at the Information Directorate. VTC was employed to conduct QoS testing on the DIMES system. DIMES, an AIMS, is described in section 5.

The testing methodology and results for latency and jitter are discussed in section 6. Concluding remarks are made in section 7 and future work in section 8.

2 Agile Information Management Systems (AIMS)

In short, an agile information management system is one that provides the information sharing and collaboration services that make an agile force possible. Information systems make up only one set of variables that agile organizations are dependent on and information management systems are a subset of that. This may seem insignificant, but that is a false presumption. The ability to manage information effectively and efficiently is the most important focus of command and control (C2) [3]. Since the information management system is an integral part of a C2 system, any deficiencies that it has will negatively affect the force that it supports. To ensure that the agile force is not affected, the information management system must exhibit the same qualities as an agile force.

According to David S. Alberts and Richard E. Hayes, "agility is arguably one of the most important characteristics of successful Information Age Organizations" [4, pp.123]. There are six dimensions to agility: robustness, resilience, responsiveness, flexibility, innovation and adaptation [4, pp.128]. An agile information management system is one that, in the least, supports all six dimensions. Each dimension and its role in an AIMS is discussed in the following subsections.

2.1 Robustness

In its application to C2 systems and forces, "robustness is the ability to retain a level of effectiveness across a range of missions that span the spectrum of conflict, operating environments, and/or circumstances" [4, pp. 128]. Robustness is therefore a measure of a system’s or force’s capability to be used in many different scenarios and applications including ones that have not yet been defined or envisioned. This differs slightly from the definition provided for C2 systems and forces. The reasoning behind this is that information systems are not as flexible as people. Once deployed, they will most likely remain constant for its life. Provided with this constraint, the systems must be future-ready.

To satisfy this requirement, the information management system must not impose any non-standard technologies, i.e. protocols and platforms, on the user. These information management systems must also be expressive enough so that it is user-extensible. In other words, the user should be able to specify new kinds of information that the management system can act upon. This way, the information management system is ready for the types of information that might arise in the future.

2.2 Resilience

"Resilience is the ability to recover from or adjust to misfortune, damage, or a destabilizing perturbation in the environment" [4, pp. 135]. For information management systems this definition is refined into the ability to recover or adjust in a timely manner. Resilience can therefore be translated into fault-tolerance and fast recovery.

When any kind of failure occurs, the system must continue to run and operate on the information that was not affected by the failure. This requirement limits the types of information management systems to either distributed, so that it can leverage fault tolerant architectures with zero down time, or replicated where the down time is measured as the time difference between when the primary system went offline and when the backup system comes online.

2.3 Responsiveness

"[Responsiveness] refers to the ability of an operating concept, C2 system, or force to act (or react) effectively in a timely manner" [4, pp. 139]. The sole purpose of an information management system is to provide information sharing and collaboration services. The action is therefore to deliver the information to the right user at the right time. In essence, the responsiveness of the system is determined by the latency introduced by the system itself in delivering a piece of information to the right user.

It is important to limit the measure to only the latency of the system and not include the latency incurred by the communication network between a user and the system. The focus is for an information management system and not for an information management infrastructure. Furthermore, it is important to understand that the requirement of acting effectively is applied under all levels of system load. It is insufficient for an information management system to be responsive under zero or little load and unable to deliver information on time under full load.

2.4 Flexibility

Flexibility is an attribute that mainly depends on cognitive abilities. At first look, it might seem that flexibility can not play a part in information management systems unless that system is a cognitive system. This is not necessarily true. By definition, "flexibility refers to the capability to achieve success in different ways" [4, pp. 143]. It is characterized by the ability to first understand changes in the battlespace and then perceive different possible futures and select the appropriate course of action [4, pp. 147].

The information management system's equivalent of the battlespace is the system state. The equivalent of different possible futures and courses of action are different possible system states and the system configurations that achieve those states. For information management systems, it is easy to select the best future, the one with the most stable system state. In most cases the most stable state is the one where the system is under the least load possible. In a distributed system, the most stable state is the one where the load is balanced across different sites. Therefore, flexibility for information management systems refers to the ability to effectively balance and manage load.

Also, flexibility in agile forces is highly depended on the ability to communicate and collaborate. This induces the requirement that the AIMS must provide effective collaboration technologies to its users.

2.5 Innovation

"Innovation is the ability to do things in new ways or to understand new things to do, particularly new ways to achieve desired ends. This involves the ability to learn over time about missions and operational environments and to take advantage of the lessons learned to create and maintain competitive advantages" [4, pp.149]. There is another quality of innovation, unpredictability. Provided that information management systems are deterministic, the unpredictability aspect of innovation cannot be satisfied. Though this is true, an information management system is still able to learn and then utilize what it learned to make improvements.

One area of research that applies machine learning techniques to information is information integration. Although the field of information integration is mostly applied to the web, its solutions and ideas can also be applied to information management. For example, Active Atlas is a system developed at the University of Southern California that uses machine learning techniques to learn how different sources in its operational environment, the web, represent data [5]. Once this knowledge is obtained, it can then generate weights for string transformations so that the different source’s information can be compared. In the end, the system will be able to recognize that “W. Main St.,” “Main St. West,” “West Main St.,” “West Main Street” and etc. all represent the same information. This technology can be applied to information management systems so that users can express interest for one kind of information and receive objects from multiple sources without having to explicitly use the data representation standards of all the different sources. A system that is capable of this is innovative.

Unfortunately integration of the above mentioned and similar techniques into information management systems are still being researched. Until they are mature enough, a middle ground, semi-innovation, needs to be defined. For the time being, an information management system can be called innovative if it provides the user the ability to access multiple sources using one single representation. It is not yet necessary for the information management system to learn how this is done automatically. Human intervention is allowed.

2.6 Adaptation

"Adaptation is the ability to alter force organization and work processes when necessary as the situation and/or environment changes" [4, pp. 153]. This implies that the system must be scalable and reconfigurable. System scalability means that the system can be expanded when the number of users (need for capacity) or the amount of information being managed (need for throughput) increases and shrunk when the need decreases. A reconfigurable system is one where the system parameters can be changed to match the application so that the system is used efficiently.

3 Quality of Service (QoS) Metrics for AIMS

3.1 Background

Quality of service metrics or parameters have traditionally been used to describe the characteristics of telecommunication networks. Jens Grabowski and Thomas Walker define QoS as "a set of parameters that characterize a connection between communication entities across a network" [6]. The metrics are categorized into: performance (e.g. latency), reliability (e.g. fault-tolerance,) and miscellaneous (e.g. security).

Furthermore, the metrics can be separated into five different types of QoS guarantees [6]. Best effort QoS values are ones where the service provider tries its best to provide the agreed upon level of service with the user. If the need arises, agreed upon QoS values can be temporarily dropped. Guaranteed QoS values, the ones that are of interest, are ones where the provider either accepts or rejects the user's requirement. Once accepted, these values can never change and are absolutely guaranteed to be met until the user finally disconnects.

Due to the fact that the next three types allow the server to disconnect users when the level of QoS is not met and also allow QoS values to be strengthened when possible, they require active monitoring of the communications channel. Compulsory QoS values are similar to best effort values. The only difference is that the user's connection is aborted if the agreed upon values are not met. Threshold QoS values take a different approach to unsatisfied QoS values than compulsory. Instead of aborting a user automatically, the service provider informs the user of the situation and the user chooses the appropriate course of action. Finally there exist the possibility of mixed threshold and compulsory QoS values.

These classifications are still applicable today, except the applicable domain has been broadened to include both network and end-systems, which includes all of the players involved between one end user to another end user [2], [7]. Information management systems consist of only one of the many layers that make up the end-to-end architecture.

Though it is important to address QoS issues found in all layers, the focus here is on information management systems.

3.2 AIMS as Real-time Multimedia Systems

Agile information management systems need to be able to support many different types of users. The types of users can range from low-tempo forces, e.g. traditional submarine warfare that can operate in days or weeks, to high-tempo forces, e.g. close air support that require response times of minutes, to real-time users, e.g. those who wish to collaborate through video teleconferencing [4, pp.139]. Although there is a large range in the required response times and performance characteristics of AIMS, the systems should always be able to satisfy the most stringent of requirements, real-time collaboration. For this reason, AIMS can be categorized as real-time systems.

Since "video teleconferencing is the most elaborate alternative [to face-to-face collaboration]" [8, pp.188], AIMS should support video teleconferencing. As a result, AIMS can then be further described as a real-time multimedia system. Naturally, AIMS inherit most of the desired quality of service attributes of real-time multimedia systems.

Quality of Service and its application to distributed multimedia systems can be defined as "the set of those technical and other parameters of a distributed multimedia system, which influence the presentation of multimedia data to the user" [2]. The parameters that affect the quality of a multimedia stream are latency and jitter.

The additional parameters of throughput, capacity, scalability, fault-tolerance and load balancing are also required for all agile information systems as discussed above. Finally, as in all information systems, security should be one of the top priorities, and therefore security is another important quality of service.

4 Distributed Information Management Enterprise Service

The Distributed Information Management Enterprise Service is an information management system under development at AFRL IF. It is a parallel and distributed instantiation of the Information Management Enterprise Service Reference Implementation also under development at AFRL IF.

4.1 Information Management Enterprise Service

The Information Management Enterprise Service (IMES) is an information management system based on the publish/subscribe distributed systems paradigm. It provides a means for a Community of Interest (COI) to share information in a centralized and organized fashion. It can be deployed as a COI service on the GIG or as a stand alone information management service. The concept of IMES is to provide users with a collection of core services that are common and consistent across all implementations. Currently the core services include, publish, subscribe, query, archive, access control and information transform.

The cornerstone of IMES is the open Common Applications Programming Interface (CAPI). The CAPI is a continually evolving open standard that defines how clients communicate with the IMES server and how components within the IMES server communicate with each other. With the CAPI, IMES can be considered a reference implementation or platform since developers are encouraged to change the actual implementation of each client or core service to tailor IMES to their application. The only requirement is that the new instantiation remain CAPI compatible. The CAPI interface enables modularity and extensibility in IMES as well as an assurance that CAPI clients and core services are interoperable.

The publish/subscribe nature of IMES ensures that the producers and consumers of information are complete decoupled. Producers create information and send such information to the server. Consumers simply register interest in specific kinds of information with the IMES server. Once the server receives any information that satisfies the consumer's description, that information is sent to the consumer. This architecture does not require any producers and consumers to communicate directly with each other. This relieves the different parties from having to use the same protocols. It also prevents bottlenecks from being formed when one producer is sending information to or one consumer is receiving information from multiple other parties.

There are three main groups of IMES users, publishers (producers of information), subscribers (consumers of current information), and query users (consumers of archived or past information). These users share information using data primitives called Information Objects (IOs). It is the CAPI and the use of user defined IOs that make IMES and its instantiations robust.

The IMES server is made up of four major components as shown in Figure 1. The publisher interface interacts with producers of information while the subscriber and query interfaces interact with information consumers. The broker ensures that the right consumers receive the right information. The repository, which is associated with the broker, is used to persist information objects. Finally, role-based access control is used to enforce user access rules. These components are used to provide the publish, subscribe, archive, query and access control core services. The information transform core service is provided by fuselets.

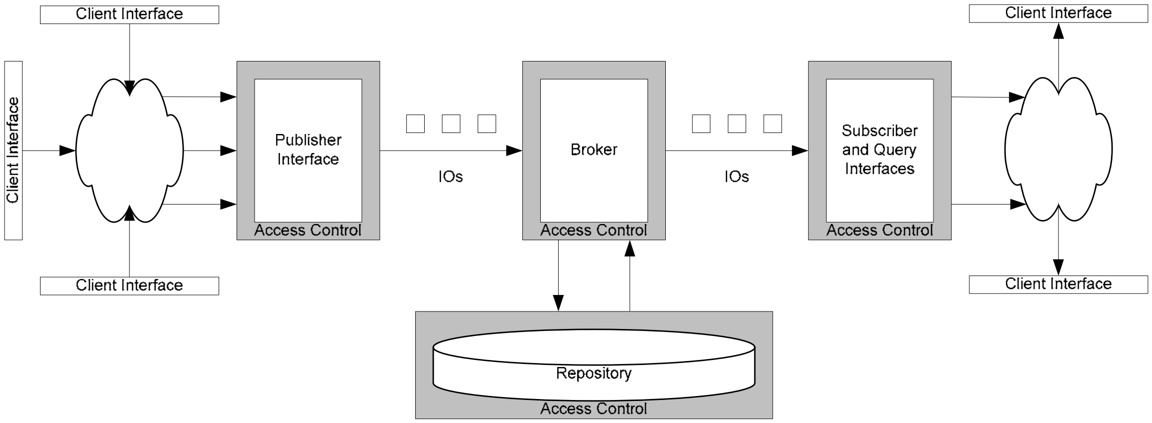


Figure 1. IMES System Overview

4.1.1 Information Objects

Information Objects consist of two components, *metadata* and *payload*. The *metadata* is an XML document that either contains all of the information or contains some of the information along with a description of the *payload*. The latter of which is required if the information being shared does not have a textual or XML representation. The *payload* is preformatted data and the exact format of the *payload* must be agreed upon by the users and noted in the *metadata* section.

IOs are analogous to well defined emails. Emails, like IOs, can also be separated into two main parts, the message body, which is text, and attachments, which is preformatted data. The main difference is that email messages do not have a structured format that must be followed.

The format of an IO's *metadata* is defined in an XML Schema file. The file defines a strict structure that the *metadata* must follow and all user-defined IOs extend from a base information object metadata. The IMES server is set up so that, before it will manage a specific type of IO, the IO's, type, version and *metadata* schema must be registered with the server. This supports user extensibility since new types of information can be managed by IMES as long as the information can be formatted into an information object.

4.1.2 Semi-innovation Through Fuselets

Fuselets facilitate semi-innovation and is used to support the information transformation core service. Fuselets are considered light weight IMES clients that are registered with the IMES server so that users can discover the fuselets available for use. They are special purpose processing clients that are supplied to users by the IMES server itself. To achieve low latency and high throughput access to information objects, fuselets are usually collocated with the IMES server.

The purpose of fuselets is to take information objects, process and manipulate them to produce new information objects that are useful to clients, which is the information transformation core service. It can be seen that if the fuselet takes data from disparate sources with different information object types and representations and translate them into a single information object type, then a client accessing the resultant IO type would be accessing information from multiple sources by using only one representation, satisfying the requirement for innovation.

4.2 Distributed Information Management Enterprise Service

The Distributed Information Management Enterprise Service is an implementation of IMES, i.e. DIMES implements the IMES core services differently but remains CAPI compatible. The purpose of DIMES is to improve the throughput and latency of IMES, as well as add data replication, load balancing and fault tolerance functionality to IMES. This is done through code and design optimizations, parallelization and distribution of the components that support the core services.

DIMES is designed for performance and scalability, which is mainly achieved through a parallelization and distribution strategy. The DIMES system architecture is shown in Figure 2. In the parallel pipeline model, all of the components in Figure 1 have been redesigned and parallelized. The Publisher Catchers are the equivalent of the Publisher Interface in IMES and the Disseminators are the equivalent of the Subscriber and Query Interfaces. What is not depicted in Figure 2 are the distributed repositories that are associated with the brokers.

This parallel pipeline model makes DIMES extremely versatile. The number of Publisher Catchers, Brokers and Disseminators can be scaled, mixed and matched to create a DIMES server that fits the desired requirements, making it adaptive. DIMES is also designed for flexibility and resilience. The Connector component in Figure 2 is used to send information objects to other DIMES servers that are part of the same federation of IMES. The Connector is designed to replicate information objects upon request by the publisher.

In times when information objects are replicated, there will be copies of the same information object located in DIMES servers in different locations. Taking advantage of this, a fault tolerant structure for clients was created to ensure that producers and consumers will always be able to use the services provided by the federation of DIMES. If for some reason a publisher cannot communicate with its nearest DIMES server, it can always communicate with other DIMES servers in the federation and can be assured that their information will be properly managed. On the subscriber and query side, up to three redundant connections can be made to different DIMES servers to ensure that if one server fails, the subscriber will continue receive information objects that match their requirements from the others.

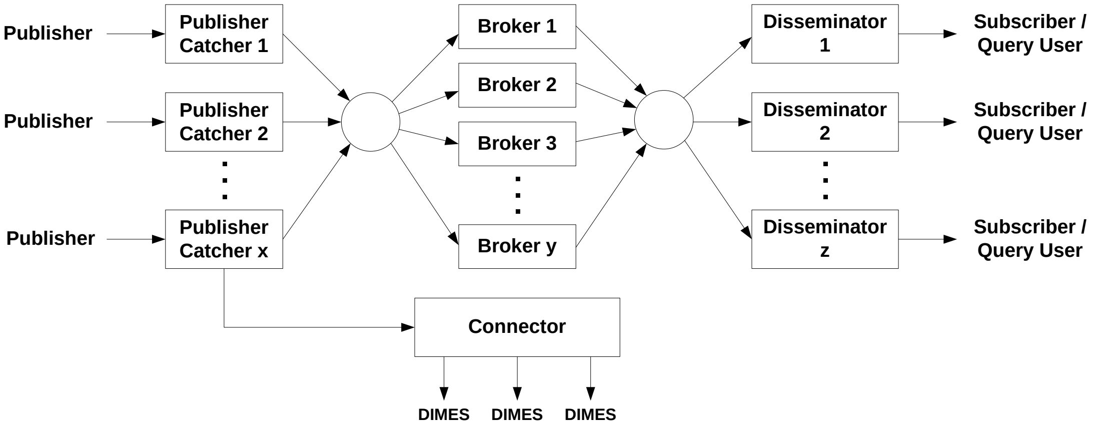


Figure 2. Parallel Pipeline Model

5 Video Test Client

5.1 Design

Video Test Client (VTC) is a flexible Windows based audio/video test platform implemented in C++. The architecture overview is depicted in Figure 3. It consists of four major components, the capturer, transmitter, receiver and player and two minor components the AVI file reader and writer. The capturer captures audio and video and encodes it into frames whereas its alternative, the AVI file reader, reads pre-encoded audio and video frames from a file. The frames are then passed onto the transmitter which then sends the information through the information management system under test. The receiver receives the frames from the information management system and passes it along to the player which renders the audio and video frames or to the AVI file writer which saves the frames for future use such as repeating tests.

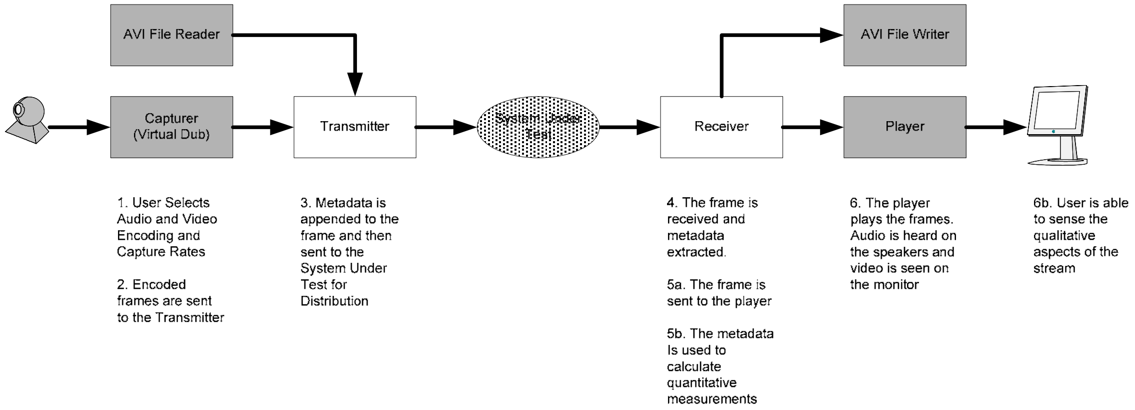

AVI File Reader
Capturer (Virtual Dub)
Transmitter
System Under Test
Receiver
AVI File Writer
Player
1. User Selects Audio and Video Encoding and Capture Rates
2. Encoded frames are sent to the Transmitter
3. Metadata is appended to the frame and then sent to the System Under Test for Distribution
4. The frame is received and metadata extracted.
5a. The frame is sent to the player
5b. The metadata Is used to calculate quantitative measurements
6. The player plays the frames. Audio is heard on the speakers and video is seen on the monitor
6b. User is able to sense the qualitative aspects of the stream

Figure 3. Video Test Client Overview

5.1.1 Capturer

The capturer component is a modified version of the versatile open source video capture and encoding tool, VirtualDub version 1.6.10. It supports any encoder that can register with DirectX. This allows the user to choose from a slew of codecs and video capture rates so that the resulting information can vary in size, quality and other characteristics. Originally VirtualDub captures audio at the suggested default rate of 2 per second (500ms each frame). Since VTC is a test platform, the audio capture rate needs to be selectable just like the video rate. Consequently, VirtualDub was modified to provide this functionality even though the audio quality tends to drop as the capture rate increases [9].

VirtualDub was designed to capture and encode audio and video and then output the results into a file in the AVI 2.0 format. Since the frames are to be passed along to the transmitter for sending, writing the frames to the file is inefficient. Therefore, VirtualDub was also modified to give the user the option to choose to write the frames into a named pipe instead of a file.

The use of the AVI 2.0 format is another issue with VirtualDub. Two of the main reasons why the AVI 2.0 format was created were to provide faster indexing than and overcome the two gigabyte file size limit of the original AVI 1.0 format [10], [11]. Since VTC streams the frames instead of simply storing them for playback, indexing and the file size limit do not pose problems. As a result, VTC only requires minimal AVI 1.0 compatibility to function and VirtualDub was modified to output the frames in an AVI 1.0 format when used with a named pipe.

5.1.2 Transmitter

The transmitter is implemented as an interface that receives the encoded frames and related metadata from the capturer. It is up to the end user to implement this interface so that the frames will be sent to the information management system.

Aside from providing the ability to send frames to the information management system for distribution, the transmitter can also append additional test information onto the stream (only if the information management system itself does not support this additional metadata, otherwise its unneeded overhead). This is done by using junk chunks defined in the AVI format. Junk chunks are normally used to enforce data alignment but have also been used for proprietary extensions to the AVI format. Proper AVI de-multiplexers simply skip over the junk chunks and read in the next valid list or chunk.

VTC encapsulates stream related metadata into the junk chunks as shown in Figure 4 to append additional information while maintaining AVI compatibility. Table 1 shows all of the currently supported extensions.

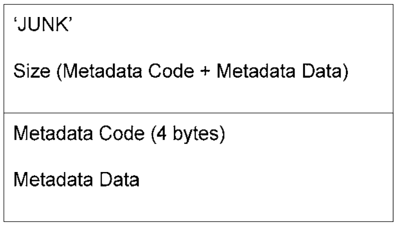


Figure 4. Metadata Extension

Metadata Code	Metadata Data
'SEND'	Time when frame is sent
'RECV'	Time when frame was received
'FRMN'	Frame Number

Table 1. Metadata Extension Definitions

5.1.3 Receiver

Like the transmitter, the receiver is an interface that needs to be implemented so it can communicate with the information management system under test. Upon receipt of a frame, it will send the encoded audio or video to the player and extract any additional information appended by the transmitter. With this additional information, the latency and jitter quality of service metrics can be calculated and logged.

The receiver also completes preprocessing of the audio and video frames. This includes re-ordering any frames that might have arrived in order and controlling which frames are sent to the player, e.g. dropping frames beyond a specific threshold.

5.1.4 Player

The player is a DirectShow based audio and video player. As long as the decoder that is associated with the encoder used by the capturer is registered with DirectX, the player will be able to render the frames passed onto it by the transmitter. To ensure that the user has as much control over the functionality of VTC as possible, the player is designed to play what is available to it without any additional processing. Frame control, e.g. pre-buffering, is completed by the receiver.

5.2 DIMESVTC

The Distributed Information Management Enterprise Service Video Test Client (DIMESVTC) is a test tool built on top of the VTC test platform. The transmitter and receiver components were implemented so that VTC is able to communicate with DIMES.

The DIMESVTC transmitter is implemented so that two separate connections are made to the DIMES server, one for audio and one for video. This allows the tester to temporarily suspend transmission of a stream, e.g. muted audio stream, without affecting the other. Furthermore, separate connections are needed since the audio and video streams have different QoS requirements.

The DIMESVTC receiver is implemented so that it can receive audio and video streams from the DIMES server. Upon receipt of a frame, the player extracts any extra information that was appended by the transmitter and logs it. It then rearranges the frames if they are out of order. Then it passes the frames along to the player.

6 DIMES Tests and Results

All of the QoS metrics identified in section 3, except security, can be tested with VTC. The throughput of the system under test can be measure with respect to either the aggregate bandwidth, the number of information objects per second or the number of simultaneous collaboration sessions. The capacity can be measured with respect to the number of simultaneous users of the system before the system is no longer able to satisfy the QoS needs of its users. These tests can then be repeated for different configurations to determine the scalability and load balancing capabilities of the system. All of these tests take advantage of the quantitative measurement capabilities of VTC.

Tests that can benefit from the qualitative nature of VTC are the ones for the fault tolerance, latency and jitter metrics. A fault tolerant system should not drop collaborative sessions or if is a master/backup system then the delay between failure and recovery should not be too long therefore the audio and video quality during a controlled system shutdown is a good indication of the system's qualities.

Due to limitations in the current versions of DIMESVTC and DIMES, only the latency and jitter results are presented.

6.1 Latency and Jitter Test

As with any test, the testing environment itself must be well controlled to ensure that the measurements and results are valid. Latency and jitter are sensitive tests where effects can be experienced from anywhere along the path from capture at the producer end to playback at the consumer end. A tester can choose to host the client on to the same machine(s) as the server, but unless the system was designed to operate in such a fashion or it can be guaranteed to the tester that the presence of the client will not affect the performance of the server, it is an unrealistic scenario. The best alternative, then, is to place the client and server on different machines but connect them through a tightly controlled, high bandwidth and low latency network to reduce the effects of the transmission as much as possible. Although in most cases, separating the effects due to the server and those due to the network is difficult.

Fortunately, DIMES provides an information object time-stamping feature that stamps the information when it enters and exits from the Publisher Catcher and Disseminator stages. Provided with these times, the server latency and jitter can be calculated with little error. Due to this ability, the interconnection between the client and the server no longer needs to be on a specialized network and a campus area network was used instead

Furthermore, the sending and receiving clients were collocated on the same machine so that the timestamps are from the same clock. The same applies to the Publisher Catcher and Disseminator.

The following is test configuration used:

- 1.2GHz laptop acting as the publisher and subscriber of audio and video information
- Dual 3.2GHz Xeon with 8GB RAM server hosting a DIMES configured with a single Publisher Catcher, a single Broker and a single Disseminator
- 100Mbps campus area network connecting the client and server
- 320x240 resolution video captured through a USB webcam at 30 fps encoded with DIVX 6.0 at a bitrate of 200kbps, keyframe interval of 300 and bidirectional encoding disabled
- 22kHz, 16bit mono audio captured through the built in microphone at 20 fps encoded with MP3 at a constant bitrate of 32kbps

6.2 Results

Figure 5, Figure 6 and Figure 7 are scatter plots of a two minute, six thousand frames, segment of data found towards the end of a data collection session. Figure 5 shows the server latency values. These values are measured as the time when the IO leaves the Disseminator subtracted by the time when it enters the Publisher Catcher. These are therefore pure processing values that do not take into account the network stack or transmission latencies. Figure 6 shows the amount of jitter found in the server according to the formula presented in RFC3550 [12]. Figure 7 shows the total latency values experienced by the IO from the time it is sent to the DIMES client publisher interface to the time it arrives at the DIMESVTC receiver from the DIMES client subscriber interface. These values include latencies contributed by the two DIMES client interfaces, the network and the DIMES server.

6.2.1 Server Measurements

There are a couple of interesting things in the server latency results. The graph for the server jitter is not as interesting since it's derived from the latency results.

First, there are numerous outlying points that appear at seemingly regular intervals. These higher latency values correspond to IOs that contain video keyframes. Looking at the interval from frame number 1000 to 2000, three high latency data points are seen. This is because the keyframe interval was set at 300, which means that there is at least one keyframe every 300 frames. This means that in the 20 seconds between frame numbers 1000 to 2000 there was very little motion in the scene.

Second, there is a rising trend in the latencies between the high data points. It appears as though the server latency drops after a high latency point and then slowly rises as more IOs arrive. An exact explanation cannot be found at the current time since more experimentation and analysis must be conducted first.

The hypothesis is that due to the use of queues, if the input rate is higher than the output rate then the queue will begin to fill up and therefore the overall server latency slowly increases. This is a possible explanation since even though the audio and video frames are set at periodic rates of 20fps and 30fps respectively totally 50fps, the actual transmission of frames does not occur at regular intervals. This is expected, especially for video frames, because different frames will require different amounts of processing. As a result, there are times when the publication rate is actually higher than the intended 50 frames a second. Then, due to the fact that the keyframes are much larger than other frames, they require a longer time to process and transmit over the network, a time slice that contains the keyframe will therefore result in a lower input rate into the queues. This can explain why there are sudden drops in server latencies after the keyframes, generating a saw-tooth pattern. Once again, this is speculation and more tests and analysis must be completed in order to arrive at the right conclusion.

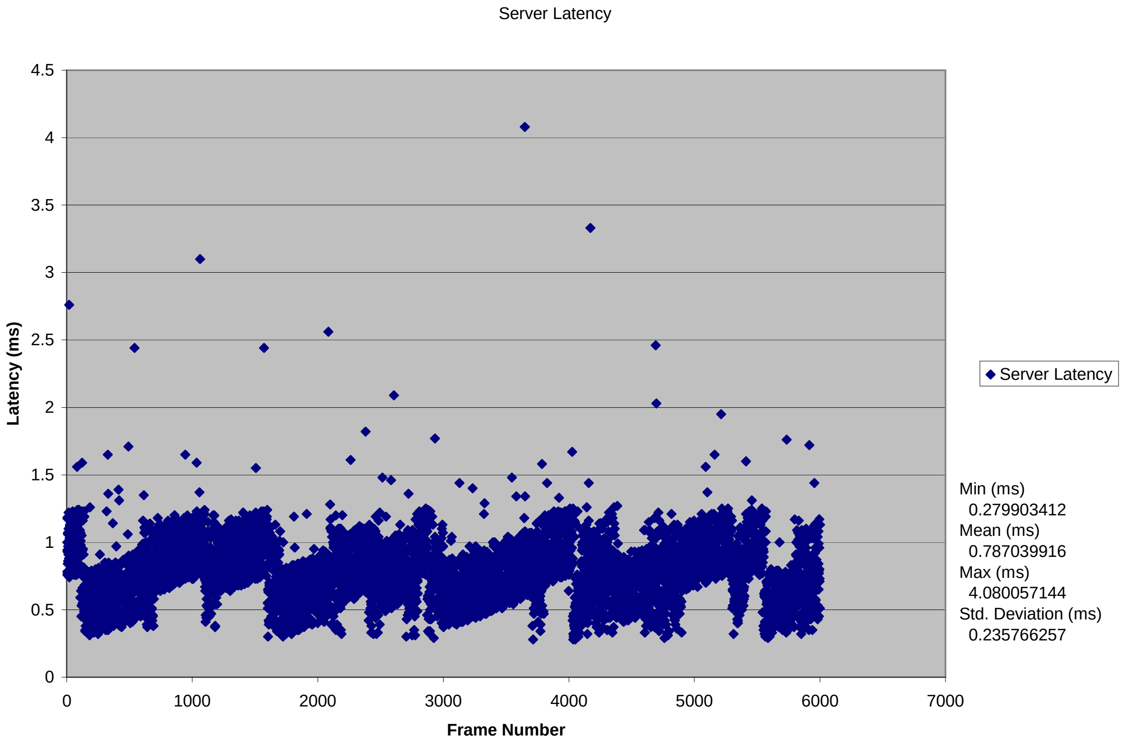


Figure 5. Server Latency

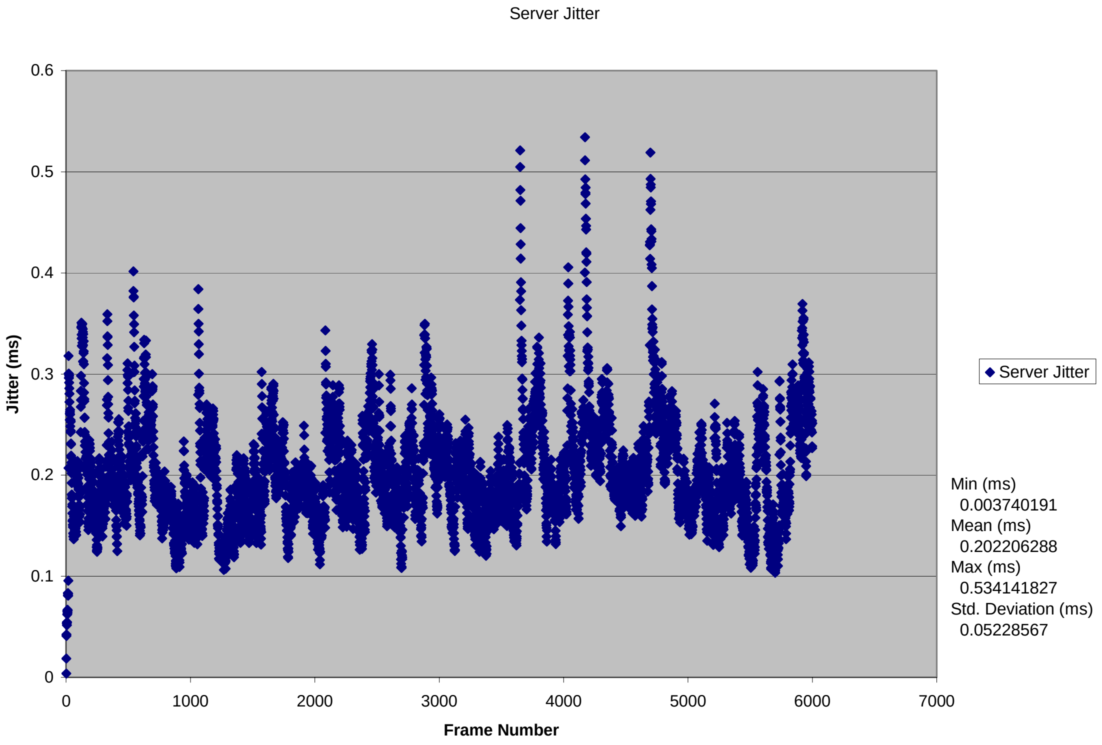


Figure 6. Server Jitter

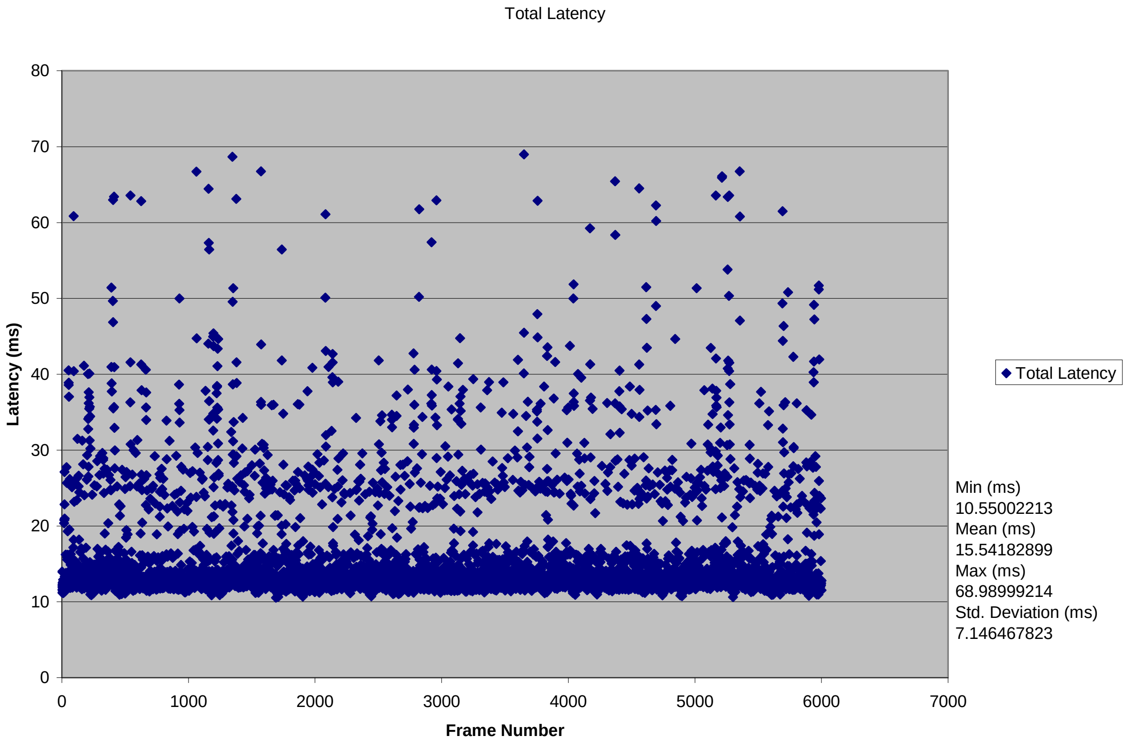


Figure 7. End to End Latency

6.2.2 End-to-end Latency

The total latency plot has an interesting aspect to it as well. There are essentially two different groups of data points, the ones below 20ms and the ones above 20ms. It turns out that most of the data between the 20ms line are audio frames. Unlike video frames that differ in size depending on the amount of change in the scene, the audio frames have an almost uniform size. Given that the audio bitrate chosen was 32kbps at 20fps then if everything was perfect, each frame would be 200 bytes. For such small amount of data, the latency due to the transmission is very close to the transmission latency itself. The data shows that there exists about an 11-12ms round trip transmission latency from the client to the server. The audio and video quality did not degrade at all and was good overall, which means that the quantitative and qualitative results are consistent.

7 Concluding Remarks

The concept of an Agile Information Management System was introduced. In particular, an AIMS is an information management system that satisfies the agile force requirements of robustness, resilience, responsiveness, flexibility, innovation and adaptation. Quality of Service metrics that stem from these six requirements were also introduced. They are latency, jitter, throughput, capacity, scalability, fault tolerance, load balancing and security.

The IMES and DIMES systems were described as AIMS. The Video Test Client was also described as well as the latency and jitter results for a 1 Publisher Catcher, 1 Broker and 1 Disseminator DIMES server through a campus area network.

The difference in the total latency and the server latency charts emphasize the advantage of having server side time-stamping capabilities. Without it, the saw-tooth nature of the server latency would have been lost inside the total latency graph, especially when the latency values are an order of magnitude less. Furthermore, the existence of the saw-tooth pattern is attributed to the fact that live audio and video data was used for testing, an advantage of using VTC.

Although the jitter chart shows large spikes whenever there is a keyframe, the maximum value of 534µs is great since 20ms is the maximum allowable value for "good" quality interaction [13].

These preliminary results show that using average value of the server latency, 787µs, the DIMES server is able to handle approximately 190 simultaneous streams within the allotted 150ms for good quality audio and video collaboration. On the other hand using the average jitter value of 202µs and the maximum allowable value of 20ms for good quality collaboration, the server is capable of handling approximately 98 simultaneous streams. As a result, for overall good quality collaboration, the single Publisher Catcher, single Broker, and single Disseminator DIMES server is able to handle a maximum of 98 simultaneous audio/video streams on average. More tests with higher load need to be completed to determine if these values persist.

8 Future Work

A conclusion about the fitness of DIMES as an AIMS cannot be made at the current time. Only the latency and jitter results for a single stream were presented. Furthermore, the data presented has raised questions concerning the way that the DIMES server operates. The saw-tooth like nature of the server latency is intriguing. Also, the current results show that the server jitter is limiting the DIMES server from 190 possible simultaneous streams to 98. Future work will involve digging down and trying to find out the cause of the saw-tooth nature of the server latency and remediation strategies for reducing the jitter introduced by the server.

References

[1] A. Mani and A. Nagarajan. "Understanding quality of service for Web services." January 2002. [Online]. Available: http://www-128.ibm.com/developerworks/webservices/library/ws-quality.html

[2] A. Vogel, B. Kerherve, G. v. Bochmann and J. Gecsei. "Distributed Multimedia Applications and Quality of Service: A Survey," in *IEEE Multimedia*, Summer 1995, pp. 10-19.

[3] D. Alberts, J. Garstka and F. Stein. "Network Centric Warfare: Developing and Leveraging Information Superiority." DoD Command and Control Research Program. Second Edition. February 2002.

[4] D. Alberts and R. Hayes. "Power to the Edge." DoD Command and Control Research Program. April 2005.

[5] S. Tejada, C. Knoblock and S. Minton. "Learning Domain-Independent String Transformation Weights for High Accuracy Object Identification." SIGKDD '02. July 2002. Alberta, Canada.

[6] J. Grabowski and T. Walter. "Testing Quality-of-Service Aspects in Multimedia Applications," in *Multimedia Applications, Proceedings of Second Workshop on Protocols for Multimedia Systems (PROMS).* Salzburg, Austria. October 1995.

[7] C. Aurrecoechea, A. Campbell and L. Hauw. "A survey of QoS Architectures," in *Multimedia Systems*. Volume 6, No. 3. pp. 138-151. 1998.

[8] D. Alberts, J. Garstka, R. Hayes and D. Signori. "Understanding Information Age Warfare." DoD Command and Control Research Program. October 2004.

[9] "Setting Audio Capture Properties." Microsoft DirectShow 9.0 Documentation. [Online] Available: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/directshow/htm/settingaudiocaptureproperties.asp

[10] "AVI RIFF File Reference." Microsoft DirectShow 9.0 Documentation. [Online] Available: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/directshow/htm/avirifffilereference.asp

[11] OpenDML AVI M-JPEG File Format Subcommittee. "OpenDML AVI File Format Extensions." September 1997. [Online] Available: http://www.the-labs.com/Video/odmlff2-avidef.pdf

[12] Network Working Group. "RTP: A Transport Protocol for Real-Time Applications." July 2003. [Online] Available: http://www.ietf.org/rfc/rfc3550.txt

[13] P. Calyam, M. Sridharan, W. Mandrawa and P. Schopis. "Performance Measurement and Analysis of H.323 Traffic." 2004 *Passive & Active Measurement Workshop.* Antibes Juan-les-Pins, France.

Testing Agile Information Management Systems with Video Test Client Case Study: DIMES

21 Jun 06

Lok Kwong Yan

Computer Engineer

AFRL/IFTC

Air Force Research Laboratory

Outline

- Agile Information Management Systems (AIMS)
- Distributed Information Management Enterprise Service (DIMES)
- Quality of Service
- Video Test Client

Information Management Systems and Agility

- Information management systems support forces by providing information sharing and collaboration services.

- In “Power to the Edge,” David S. Alberts and Richard E. Hayes stated that there are six dimensions to force agility: Robustness, Resilience, Responsiveness, Flexibility, Innovation and Adaptation.

- An information management system that supports the six dimensions are agile.

Robustness

- “Robustness is the ability to retain a level of effectiveness across a range of missions that span the spectrum of conflict, operating environments, and/or circumstances.” (Alberts and Hayes)

- Future Ready
  - Open standards ensures effectiveness through different operating environments
  - User extensible ensures effectiveness in different conflicts

Resilience

- “Resilience is the ability to recover from or adjust to misfortune, damage, or a destabilizing perturbation in the environment.” (Alberts and Hayes)

- Misfortune, damage and destabilizing perturbations can all be classified as “faults”
- Replicated systems can provide the capability to recover from faults
- Fault tolerant systems provide the capability to dynamically adjust to faults

Responsiveness

- “[Responsiveness] refers to the ability of an operating concept, C2 system, or force to act (or react) effectively in a timely manner.” (Alberts and Hayes)

- Timeliness is not only dependent on the system but also the infrastructure.

- Networks are less predictable and controllable than systems
  - AIMS needs to be as responsive as possible

Flexibility

- “Flexibility refers to the capability to achieve success in different ways.” (Alberts and Hayes)
- It is characterized by the ability to first understand changes in the battlespace and then perceive different possible futures and select the appropriate course of action.
- The only battlespace or space that the system knows and can control is its own system state. Therefore, it should characterize the inputs that affect the state and take effective measures to ensure a stable system state in the future.
  - Load balancing

Innovation

- “[Innovation] involves the ability to learn over time about missions and operational environments and to take advantage of the lessons learned to create and maintain competitive advantages.” (Alberts and Hayes)

- Learning is largely a cognitive process
- Learn from information about missions and environments to help better serve the users
- Information transformation
- Semi-innovation is innovation without self-learning

Adaptation

- “Adaptation is the ability to alter force organization and work processes when necessary as the situation and/or environment changes.” (Alberts and Hayes)

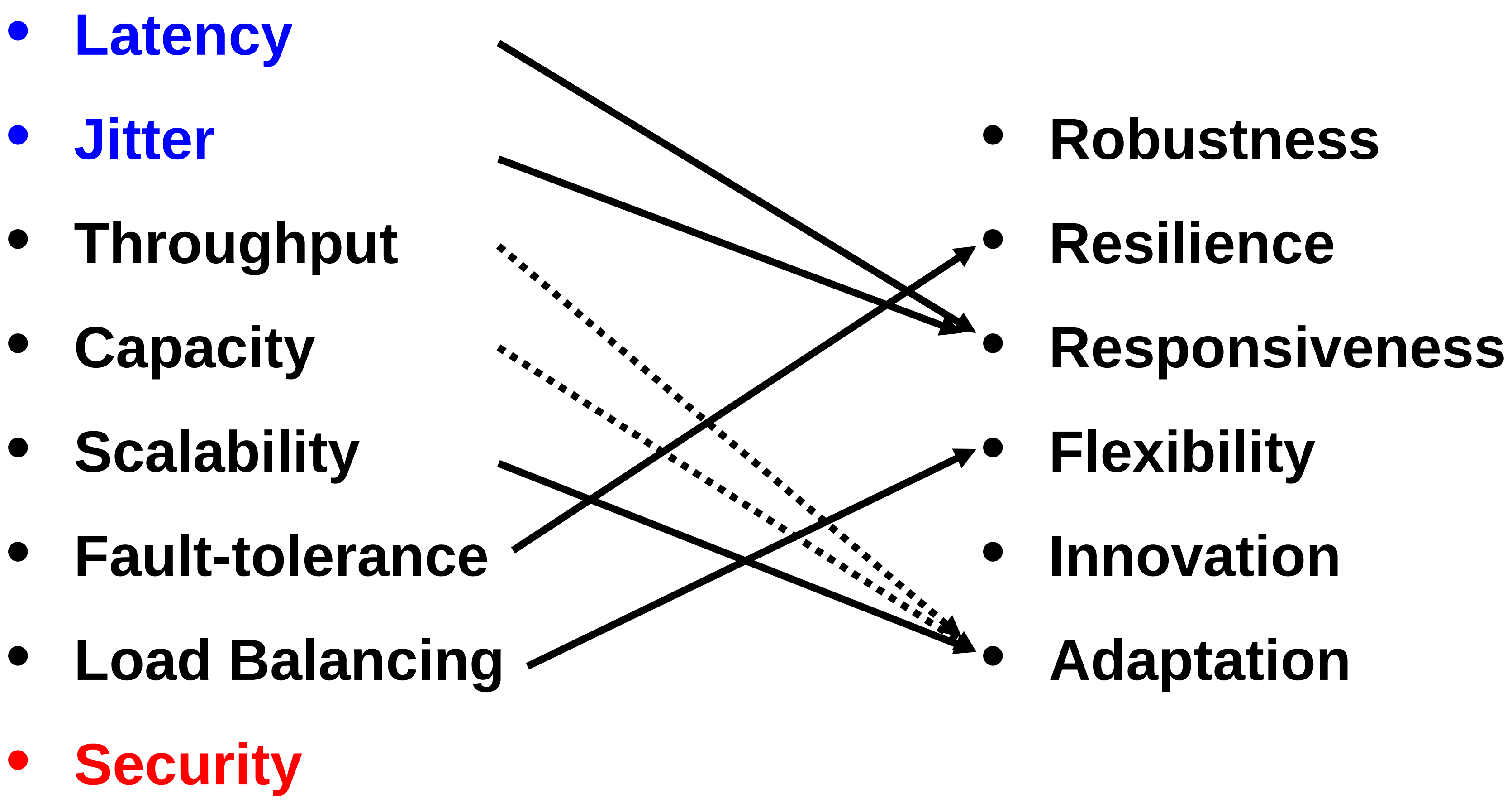
- Scalability allows users to configure a more or less capable system to match the needs
  - Capacity
  - Throughput
- Reconfigurable systems enable users to select the best configuration and algorithms for the tasks
  - Modularity

Quality of Service

- In “Testing Quality-of-Service Aspects in Multimedia Applications,” Jens Grabowski and Thomas Walker define QoS as “a set of parameters that characterize a connection between communication entities across a network.”

- The “connection” consists of
  - Clients/Users
  - Communications channel
  - Servers and services, such as the Information Management service

QoS Mappings

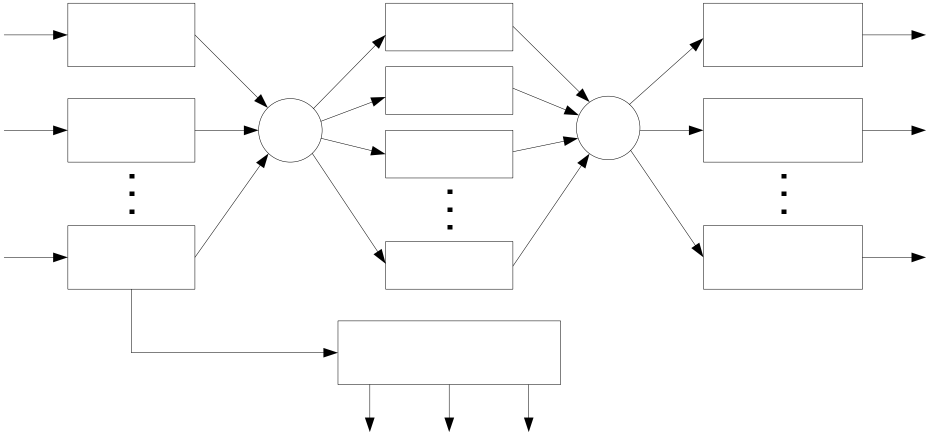


- Latency
- Jitter
- Throughput
- Capacity
- Scalability
- Fault-tolerance
- Load Balancing
- Security

- Robustness
- Resilience
- Responsiveness
- Flexibility
- Innovation
- Adaptation

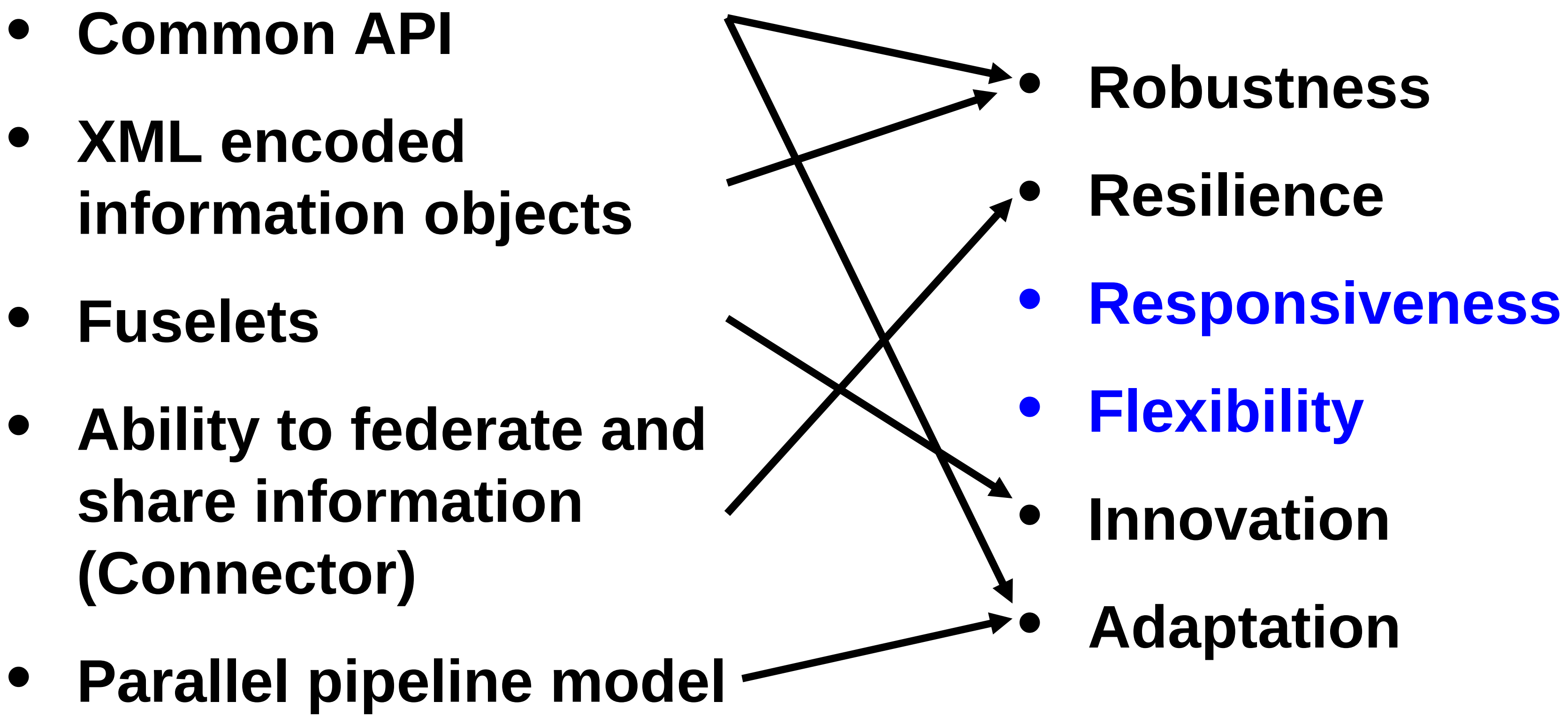
DIMES

- Developed at AFRL/IFTC
- Extends the Information Management Enterprise System
- Parallel and Distributed
- Scalable
- Low latency
- High throughput
- High capacity
- Secure
- Fault Tolerance
- Load balancing



Parallel Pipeline Model

DIMES Mapping to AIMS



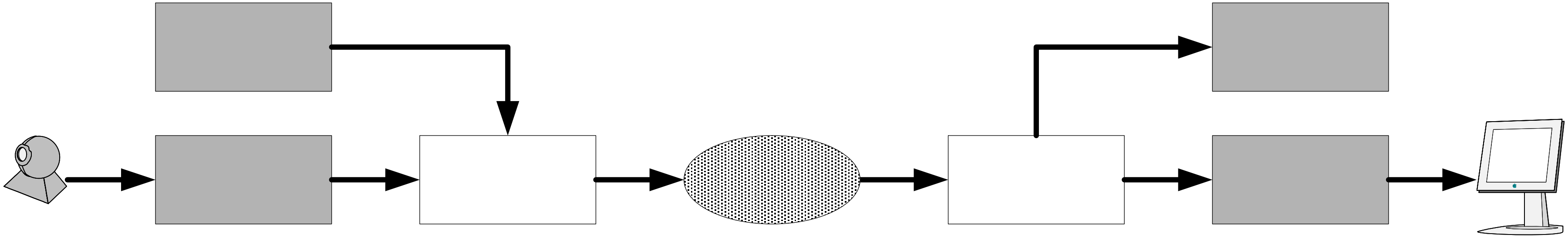
- Common API
- XML encoded information objects
- Fuselets
- Ability to federate and share information (Connector)
- Parallel pipeline model

- Robustness
- Resilience
- Responsiveness
- Flexibility
- Innovation
- Adaptation

Video Test Client

- Provides the foundation for building an audio/video collaboration tool
- Uses audio and video as the unit of information
- Provides qualitative and quantitative results
- Utilizes Microsoft DirectShow to allow users to control the rate and site of information production
  - Select and configure encoding algorithms
  - Select capture rates

Video Test Client Figure



AVI File Reader

Video Test Client Overview

Capturer

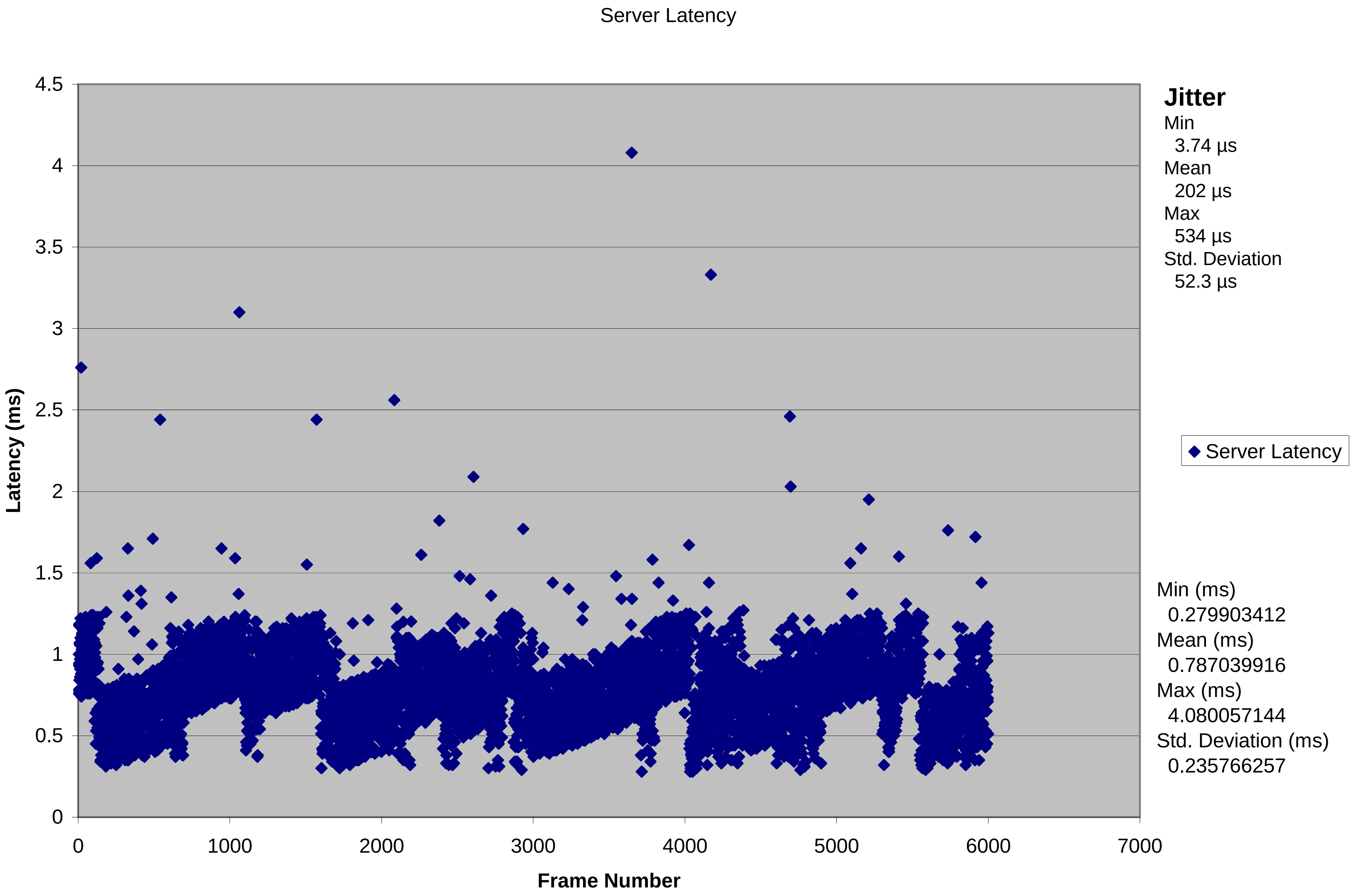
Transmitter

System

Example Setup/Configuration

- 1.2GHz laptop acting as both the publisher and subscriber of audio and video information
- Dual 3.2GHz Xeon with 8GB RAM server hosting a DIMES configured with a single Publisher Catcher, a single Broker and a single Disseminator
- 100Mbps campus area network
- 320x240 video at 30fps
  - DIVX 6.0 at 200kbps and 300 keyframe interval
- 22kHz, 16bit mono audio at 20fps
  - MP3 at 32kbps

Example Result
DIMES Server Latency



Jitter
Min
3.74 µs
Mean
202 µs
Max
534 µs
Std. Deviation
52.3 µs

Min (ms)
0.279903412
Mean (ms)
0.787039916
Max (ms)
4.080057144
Std. Deviation (ms)
0.235766257

Summary

- An agile information management system is a system that provides information sharing and collaboration services while exhibiting, robustness, resilience, responsiveness, flexibility, innovation and adaptation.
- The Distributed Information Management Enterprise System is an AIMS under development at AFRL/IFTC.
- Quality of Service metrics such as latency, jitter, throughput, capacity, scalability, fault-tolerance, load balancing and security can be used to compare AIMS.
- Video Test Client is a tool designed to assess the fitness of an AIMS in providing audio and video collaboration services and to measure QoS metrics.
  - It can measure latency and jitter out of the box.
  - It can be instrumented to measure throughput and capacity

Questions

- Now

- Later
  - Email: Lok.Kwong.Yan@rl.af.mil